Unix Experience

Sharing UNIX knowledge

Publié le **12 décembre 2011** by **Loïc BLOT**

← Précédent Suivant →

Squid: Proxy Web, Authentification, Cache et Filtrage

Introduction

Squid est un service de proxy cache web. Il permet de filtrer, mettre en cache les informations et authentifier les utilisateurs. Cette solution puissante permet de pouvoir gérer efficacement son parc et de pouvoir répondre aux attentes du code des postes et télécommunications en matière de trafic Internet.

Installation

Sous Debian:

```
aptitude install squid
```

Sous FreeBSD:

```
cd /usr/ports/www/squid31
make install clean
```

Activez tous les modules dont vous avez besoin lorsque l'invite vous le demandera.

Configuration

La configuration de SQUID est extrêmement vaste et bien documentée. Elle se trouve dans /etc/squid3/squid.conf (/usr/local/etc/squid/squid.conf.documented sous FreeBSD)

Pour inclure des fichiers il suffit d'utiliser la directive suivante.

```
include <path>
```

Je vous suggère de découper le fichier squid.conf de la manière suivante, pour plus de simplicité:

- squid.conf
- squid.acl.conf (les définitions des listes d'accès)
- squid.access.conf

DIRECTIVES GLOBALES

Port de connexion, spécifie le port et l'interface d'écoute. Par défaut toutes les adresses sur le port 3128

```
http_port <host>:<port>
```

LES LISTES D'ACCÈS

Les ACL sont des listes de contrôle, comme celles de cisco, basées sur le niveau 3 de la couche OSI. On distingue plusieurs type d'ACL:

- ACL sur une plage d'adresses sources

Exemple:

```
acl compta_service src 10.4.15.0/25
```

on définit l'ACL nommée compta_service de source 10.4.15.0/25, donc un réseau et son masque court sur une plage IP donnée.

Note: On aurait pu mettre une IP précise plutôt qu'en réseau

- ACL locale distante

Exemple:

```
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1
```

Ici on autorise dans la règle to_localhost l'ensemble de la classe A 127.0.0.0/8 et son masque inverse. Comme vous le voyez on peut également utiliser l'IPv6 (::1)

- ACL sur un port

Exemple:

```
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
```

Le fait de mettre plusieurs fois l'ACL correspond à une seule ACL avec plusieurs valeurs.

```
acl CONNECT method CONNECT
```

Cette règle est conseillée pour le TCP sinon cela risque de causer des soucis.

- ACL sur des URLs

Vous pouvez également définir un mot à chercher dans une URL.

Exemple:

```
acl blacklisted_words url_regex porn warez blowjob
```

L'ACL blacklisted_words va faire un url_regex (expression régulière) et vérifier si les mots porn warez et blowjob sont dans l'URL.

- ACL sur des domaines

On peut également bloquer des domaines (plus léger que la regex)

```
acl blacklisted_domains_cours dstdomain .facebook.com facebook.com .youtube.com
```

Cette ACL va bloquer les domaines distants facebook, youtube, dailymotion et deezer

APPLICATION DES ACLS

Prenons un exemple afin de montrer l'application des ACL.
Exemple:

```
http_access deny !sallemanip !cisco_service !admin
```

On refuse ici tous les trafics qui ne sont pas dans les ACL nommées.
Le caractère « ! » indique l'inverse d'une règle

```
http_access deny !Safe_ports
```

Ici de même on refuse les ports non autorisés

```
http_access deny CONNECT !SSL_ports
```

Et pour terminer on refuse les connexions sur un type de port donné

```
http_access deny blacklisted_domains !cisco_service
```

Cette règle va définir que l'on refuse les domaines blacklistés pour tout ce qui n'est pas matché par l'ACL cisco_service.

MODE TRANSPARENT

Pour configurer le mode transparent 2 choses sont à faire :
dans le fichier squid.conf, rajouter la directive transparent dans la directive http_port

```
http_port 3128 transparent
```

Il faut ensuite appliquer une redirection de ports vers le port de squid

Sous Debian:

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 8080 -j REDIRECT --to-port 3128
```

Sous FreeBSD, dans le fichier de configuration de PacketFilter

```
rdr on em0 inet proto tcp from any to any port 80 -> 127.0.0.1 port 3128
pass in on $int_if inet proto tcp from any to 127.0.0.1 port 3128 keep state
pass out on $ext_if inet proto tcp from any to any port www keep state
```

Lancement du service

Sous Debian:

```
service squid3 start
```

Sous FreeBSD:

```
service squid start
```

CONFIGURATION OPTIONNELLE

- Limitations de bande passante

Squid intègre un filtrage de bande passante. Il faut savoir que cela consomme du CPU. Ici nous allons appliquer une limitation de bande passante à 1Mbit/sec pour chaque utilisateur du réseau 192.168.1.0/24

```
acl only1mbusers src 192.168.1.0/24
delay_pools 1
delay_class 1 3
delay_access 1 allow only1mbusers
delay_access 1 deny all
delay_parameters 1 512000/512000 -1/-1 128000/512000
```

Je vous invite à consulter le tutoriel du wiki de squid présent ICI pour plus d'informations

- Login via LDAP

Pour vous authentifier sur un annuaire LDAP, il suffit de créer les directives suivantes dans le squid.conf

```
auth_param basic program /usr/lib/squid3/squid_ldap_auth -b "ou=people,dc=example,dc=com"
-h ldap.example.com
auth_param basic credentialsttl 3 hours
auth_param basic realm Merci de vous authentifier
auth_param basic children 5
```

Sous FreeBSD remplacer la ligne par:

```
auth_param basic program squid_ldap_auth -b "ou=people,dc=example,dc=com"
-h ldap.example.com
```

- Login via AD

On va déjà installer les dépendances winbind, samba, krb5-config, krb5-user.

```
aptitude install winbind samba krb5-config krb5-user
```

- Configuration de Samba

Ouvrez le fichier /etc/samba/smb.conf

```
[global]
   workgroup = DOMAIN
   password server = activedirectory
   realm = DOMAIN.TLD
   netbios name = squid-gateway
   server string = %h server
   dns proxy = no
   log file = /var/log/samba/log.%m
   max log size = 1000
   syslog = 0
   panic action = /usr/share/samba/panic-action %d
   security = domain
   encrypt passwords = true
   passdb backend = tdbsam
   obey pam restrictions = yes
   unix password sync = yes
   passwd program = /usr/bin/passwd %u
   passwd chat = *Entersnews*spassword:* %nn *Retypesnews*spassword:* %nn
*passwordsupdatedssuccessfully* .
   pam password change = yes
```

```
load printers = no
idmap uid = 10000-20000
idmap gid = 10000-20000
template shell = /bin/bash
template homedir = /home/%D/%U
winbind separator = +
winbind enum groups = yes
winbind enum users = yes
winbind use default domain = yes
prefered master = no
```

- Configuration de Kerberos

Ouvrez maintenant le fichier /etc/krb5.conf

```
[libdefaults]
        default_realm = DOMAIN.TLD
        kdc_timesync = 1
        ccache_type = 4
        forwardable = true
        proxiable = true

        v4_instance_resolve = false
        v4_name_convert = {
                host = {
                        rcmd = host
                        ftp = ftp
                }
                plain = {
                        something = something-else
                }
        }
        fcc-mit-ticketflags = true
[realms]
        DOMAIN.TLD = {
                kdc = activedirectory.domain.tld
                admin_server = mail@domain.tld
                default_domain = DOMAIN.tld
                a_server = activedirectory.domain.tld
        }
[domain_realm]
        .domain.tld = activedirectory.domain.tld
        domain.tld = activedirectory.domain.tld
[login]

[appdefaults]
pam = {
        debug = false
        forwardable = true
}
```

- Configuration du pam (uniquement sous Debian)

Pour terminer nous allons lier winbind au pam. Ouvrez /etc/nsswitch.conf

```
passwd:         compat winbind
group:          compat winbind
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

- Redémarrage des services

```
service winbind restart && service samba restart
```

- Joindre le domaine

```
net ads join -U administrateur
kinit -V administrateur@DOMAIN.TLD
```

- Vérification de la liaison au domaine

Pour Winbind:

```
wbinfo -u
```

Pour Kerberos:

```
klist
```

Pour Samba:

```
smbclient -L localhost -U DOMAIN\Administrateur
```

Il existe un problème de droits bien connu, pour le résoudre tapez les commandes suivantes:

```
usermod -G winbind_priv squid
```

Terminer par ceci pour relancer les pipes d'authentification NTLM

```
killall ntlm_auth
```

- Configuration de Squid

```
auth_param ntlm program /usr/bin/ntlm_auth --helper-protocol=squid-2.5-ntlmssp
auth_param ntlm children 5
```

BLOCAGE DE MSN-MESSENGER

Voici les ACL à implémenter:

```
acl msnp req_mime_type -i ^application/x-msn-messenger$
acl msnq rep_mime_type ^application/x-msn-messenger$
http_reply_access deny msnp
http_reply_access deny msnq
```

BLOCAGE DE SKYPE

On bloque l'user-agent de skype. Sur les version récentes, skype ne présente plus d'UserAgent. Aucun des navigateurs Internet récent ne naviguant sans UserAgent, on va donc bloquer ceci.

```
acl Skype_UA browser ^skype^
http_access deny Skype_UA
acl invalidUserAgent browser S+
http_access deny invalidUserAgent
```

Cet article fait partie de la catégorie **Sécurité** et a été rédigé par **Loïc BLOT**. Enregistrer le **permalien**.

Laisser un commentaire

Votre adresse de messagerie ne sera pas publiée. Les champs obligatoires sont indiqués avec *

Nom *

Adresse de contact *

Site web

Commentaire

Laisser un commentaire

Thème édité par Loïc BLOT, CNRS